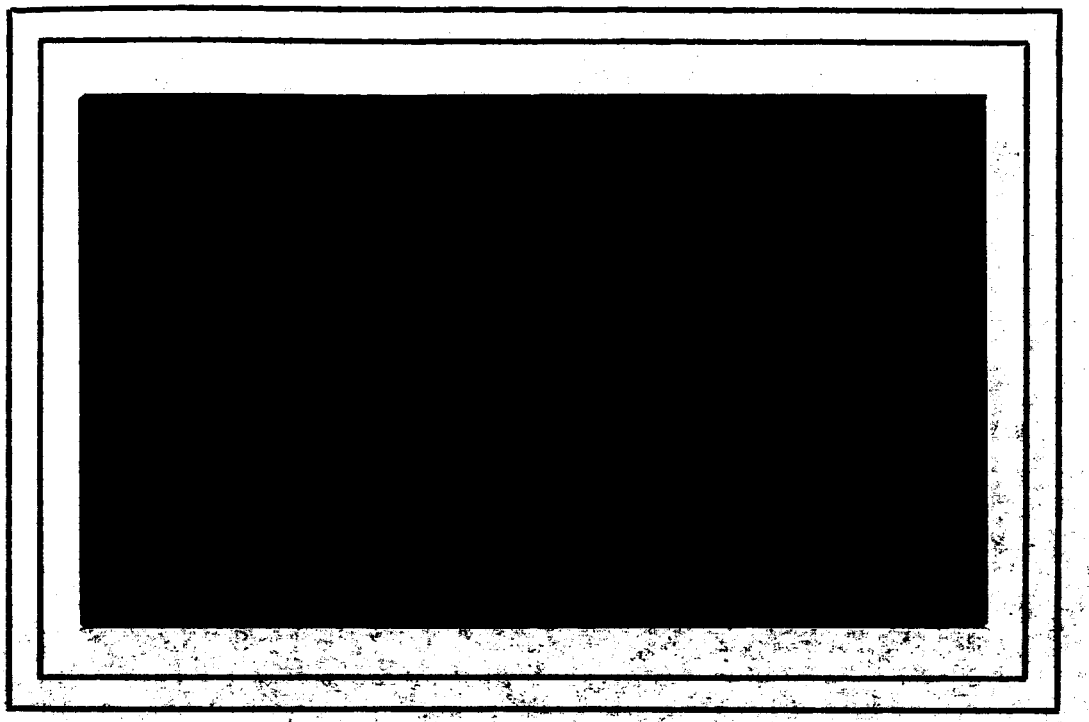

UNIVERSITY OF MARYLAND
COMPUTER SCIENCE CENTER

COLLEGE PARK, MARYLAND

Technical Note TN-65-17
NsG-398

April 1965

The University of Maryland 1401 Transliteration Preprocessor for ALGOL 60 to ALCOR-ILLINOIS 7090/94 Conversion*

by

H. D. Wactlar
Computer Science Center, University of Maryland
College Park, Maryland

*This work was supported by the National Aeronautics and Space Administration Grant NsG-398 to the Computer Science Center, University of Maryland, College Park, Maryland

ABSTRACT

This report describes a 1401 preprocessor for Flexowriter typed ALGOL programs employing an IBM 1011 paper tape reader. The preprocessor simultaneously converts from paper tape to punched cards and transliterates from reference language notation to that of the ALCOR-ILLINOIS 7090/94 translator. This eliminates hand transliteration to individual compilers thereby facilitating exchange and implementation of ALGOL programs.

The transliteration procedure is described as a model for conversion to other compilers. Schemes for increasing the character set to allow distinction between upper and lower case alphabetic characters are discussed. Conventions are described in the appendices.

TABLE OF CONTENTS

APPENDICES

The University of Maryland 1401 Transliteration Preprocessor for ALGOL 60 to ALCOR-ILLINOIS 7090/94 Conversion

I. Introduction

As explained in the original and revised Algol 60 Reports (1,2), three different levels of language are recognized in the representation of an Algol program. The restrictions imposed on the character sets by a particular computer are usually overcome by accompanying the hardware representation by a special set of rules for transliterating from the reference language. This transliteration process, however, may incur a costly interface, unnecessarily requiring the time and effort of the programmer. This, in turn, seriously hinders the dissemination of programming advances expressed in Algol.

This paper describes a project in which the transliteration process is minimized so that it may be accomplished by the typist without requiring pre-scanning by the programmer. This is achieved by use of a flexowriter on which the program is typed and a 1401 preprocessor program which produces the transliterated hardware language format on punched cards from the flexowriter paper tape.

The hardware representation used at this Center is the Alcor-Illinois 7090/94 translator. The Algol 60 to Alcor-Illinois transliteration scheme is listed in Appendix 1. Although the programs presently available do not provide for distinguishing between upper and lower case alphabetic characters, they have been designed to allow simple implemention of some of the schemes discussed in Section III.

---

1. Naur, P., et. al. Report on the Algorithmic Language ALGOL 60 Comm. ACM 5 (May 1960), 299-314

2. ____. Revised Report on the Algorithmic Language ALGOL 60. Comm. ACM 6 (Jan. 1963), 1-17.

II. Transliteration Procedure

Since most publication language representations present only minor variations from the reference language format, an attempt has been made to transliterate from the reference language character set. The limited flexowriter character set again imposes restrictions on character availability. However, the conventions followed in transliterating from the reference language to the flexowriter keyboard are of a format simple enough for a typist to follow without knowledge of their significance. The rules for flexowriter typing of reference language programs appear in Appendix 2.

The paper tape produced by the flexowriter is input through an IBM 1011 Paper Tape Reader to an 8K storage IBM 1401 computer. The computer reads one record as a basic unit to be analyzed. A record is defined as all those characters between two carriage returns. There is no significance to the appearance of a carriage return except as a definition of input record length. The carriage return may be typed within any word or sequential character code. The record (to be a maximum length of 999 flexowriter punchings) is read without word marks into a cleared area of storage.

All upper case characters are set with word marks and all carriage control indicators are set to the same unique character. The word-marked text is then scanned and all the upper case (word-marked) characters are converted to their unique representations with the word marks removed. This is accomplished by a linear search of a table containing the lower case characters and their upper case correspondences. The table is arranged in order of expected character frequency so that the majority of lower to upper case conversions are made after only three comparisons.

The new array of characters is then rescanned to compress it so that all over-typed characters (e.g. underlined words) are word-marked and all control characters (e.g. backspace, upper case) are deleted. If one of the two overtyped characters is an underline, the underlined character is word-marked; similarly, for left and right string quotes, the parenthesis is word-marked. Any other character may have either member word-marked to represent the multiple character, as each word-marked special character has a unique interpretation. Those for which there is no proper interpretation are treated as underlined characters and output according to the Alcor hardware convention, enclosed by apostrophes.

Those characters which are considered to be underlined are treated by a scanning subsection which searches for a particular end character in the string of underlined characters which corresponds to the first non-blank character of the string. In this manner we search for the ending of a particular Algol word symbol which is presumed to have been intended, judging from the initial characters of the underlined string. For example, if the underlined string begins with a "B", the string is searched for an "N", presuming that either "BEGIN" or "BOOLEAN" was intended. Upon encountering the "N",

the string is assummed terminated and any remaining underlined characters of the string are treated as an entirely separate string. Similarly, if the string begins with a "TH", a terminal "N" is sought ("THEN"?); if the string begins with a "TR", a terminal "E" is sought ("TRUE"?).

In addition to the overtyped and underlined characters, the typing conventions use certain sets of sequential characters to represent particular Algol symbols (e.g. "=> " represents "$\supset$"). The text is scanned for these particular sequences whenever the first member of a possible sequence is encountered. These prescribed sequences of characters may of course be separated by blanks and carriage returns and still result in the proper interpretation.

Once all the underlined, overtyped and sequential characters of one record have been recognized and transformed, the characters are output 72 characters at a time into the punch area and output. If the output exceeds 72 characters, the remaining characters are output on succeeding cards, each card having the same number of leading blanks as the first for that record.

The program then repeats the process, clearing storage and reading succeeding paper tape records. The processing ends when the reader runs out of paper tape. After this last record is transformed, termination messages are printed and the number of cards output is indicated.

III. Upper and Lower Case Alphabetic Distinction

There are basically two forms of solution to the problem of upper/lower case distinction of alphabetic characters. The first allows the facility by providing additional restrictions or prescanning. The second requires analysis of the program content and structure with the aim of establishing symbol tables to allow creation of replacement symbols.

Among the former schemes is the capital replacement procedure. This method replaces every occurrence of a particular upper case character with some other character, both of which are designated on one or more control cards to be read as data at the beginning of every job. Thus, if a program uses "a" as an index and "A" as a sum, but nowhere in the program uses the symbol "x" or "X", one may replace all "A"'s with "X"'s. Implementation is accomplished by adding the letters and their corresponding replacements to the table of upper case replacement characters. The change from "A" to "X" is thereby accomplished in the initial scanning of the text. It is one of the most easily implemented of schemes.

Similar procedures may either precede every "A" by an "X", automatically double every upper case character (e.g. from "A" to "AA"), or prefix every upper case character with some other character with the predetermined restriction that this latter character not appear elsewhere in the program. All of these techniques (except the last) require that the text be prescanned to determine the proper replacement character or to judge if the proposed scheme is adequate. The last procedure restricts the original program to one less alphabetic character. One thereby gains one Algol character set feature at the loss of other character set generalities.

The alternate procedures that require syntax analysis of the program content to establish symbol tables, produces difficulties which cannot be considered within the framework of the current programs. In essence, such syntax analysis on the original input strings duplicate the procedures of the basic hardware compiler. It seems unreasonable to prepare such elaborate preprocessors to accomplish the initial transliteration. The input programs are generally not of such complexity as to justify this kind of analysis.

IV. Five Channel ALCOR Convention Paper Tape Conversion

Following the conventions described in ALCOR Group Representation of ALGOL Symbols (Comm. ACM 6 (1963), 597; corrected Comm. ACM 7 (1964), 189), a 5-channel paper tape to punched card conversion preprocessor has been prepared. It assumes the same machine configuration described in Appendix 4.

There are no provisions for upper/lower case alphabetic character distinction as the standard 5-channel teletype convention precludes such multiple character representations. The only programmed restriction is that the typed members of the assignment symbol (:=) may not be separated by a carriage return.

The conventions adopted are reproduced in Appendix 5.

Appendix 1

ALGOL 60 to ALCOR - ILLINOIS 7090/7094 Transliteration Scheme

Reference Language Symbol	Hardware Representation
go to	'GO TO'
if	'IF'
then	'THEN'
else	'ELSE'
for	'FOR'
do	'DO'
step	'STEP'
until	'UNTIL'
comment	'COMMENT'
begin	'BEGIN'
end	'END'
own	'OWN'
Boolean	'BOOLEAN'
integer	'INTEGER'
real	'REAL'
array	'ARRAY'
switch	'SWITCH'
procedure	'PROCEDURE'
string	'STRING'
label	'LABEL'
value	'VALUE'

Appendix 1 (Continued)

Reference Language	Description	Hardware Representation
,	comma	,
.	decimal point	.
10	base 10	'
:	colon	..
;	semicolon	.,
:=	assignment symbol	.=
# or b	blank space	
(	left parenthesis	(
)	right parenthesis	)
[	left bracket	(/
]	right bracket	/)
‹	left string quote	'('
›	right string quote	')'
+	plus	+
-	minus	-
×	multiplication	*
/	division	/
÷	integer division	//
↑	exponentiation	'POWER'
<	less than	'LESS'
≤	less than or equal to	'NOT GREATER'
=	equal to	'EQUAL'
≥	greater than or equal to	'NOT LESS'
>	greater than	'GREATER'
≠	not equal to	'NOT EQUAL'
≡	logical equivalent	'EQUIV'
⊃	logical implies	'IMPL'
∨	logical or	'OR'
∧	logical and	'AND'
¬	logical negation	'NOT'

Appendix 2

Flexowriter Typing Conventions for ALGOL Reference Language Programs

I. Special Characters

Character	To Be Typed As
÷ (integer division)	: and - overtyped
≠ (not equal)	= and / overtyped
‘ (left string quote)	(and ' overtyped
’ (right string quote)	) and ' overtyped
∧ (logical and)	A and _ overtyped
∨ (logical or)	V and _ overtyped
¬ (logical negation)	N and _ overtyped
≦ (less than or equal to)	≤ < and _ overtyped
≧ (greater than or equal to)	≥ > and _ overtyped
× (multiplication)	*
(blank)	
⊃ (logical implication)	=> = and > following
≡ (logical equivalence)	<=> < and = and > following

II. **boldface** All boldface words are to be underlined. boldface

III. General Instructions

1. All overtyping and underlining of characters is to be accomplished by use of the backspace. The order in which the characters are to be typed is immaterial. The only criteria is that the reread tape produce the proper flexowriter typed page.

2. All tape feed and stop codes will be ignored by the processor and may therefore be typed at the discretion of the typist.

Appendix 3

IBM 1011-Flexowriter Decoding Scheme

Tape punchings 87654f321	Keyboard	Decode Exit Hubs	Encode Hubs
oo . o	a / A	A	A BA 1
oo . o	b / B	B	B BA 2
ooo . oo	c / C	C	C CBA 21
oo .o	d / D	D	D BA 4
ooo .o o	e / E	E	E CBA 4 1
ooo .oo	f / F	F	F CBA 42
oo .ooo	g / G	G	G BA 421
oo o.	h / H	H	H BA 8
oooo. o	i / I	I	I CBA 8 1
o o . o	j / J	J	J CB 1
o o . o	k / K	K	K CB 2
o . oo	l / L	L	L B 21
o o .o	m / M	M	M CB 4
o .o o	n / N	N	N B 4 1
o .oo	o / O	O	O B 42
o o .ooo	p / P	P	P CB 421
o oo.	q / Q	Q	Q CB 8
o o. o	r / R	R	R B 8 1
oo . o	s / S	S	S C A 2
o . oo	t / T	T	T A 21
oo .o	u / U	U	U C A 4
o .o o	v / V	V	V A 4 1
o .oo	w / W	W	W A 42
oo .ooo	x / X	X	X C A 421
ooo.	y / Y	Y	Y C A 8
o o. o	z / Z	Z	Z A 8 1
. o	1 / ↑	1	1 1
. o	2 / +	2	2 2
o . oo	3 / [	3	3 C 21
.o	4 /]	4	4 4
o .o o	5 / =	5	5 C 4 1
o .oo	6 / $	6	6 C 42
.ooo	7 / 10	7	7 421
o.	8 / '	8	8 8
oo. o	9 / (	9	9 CB 1
o .	0 /)	0	0 C 8 2
o .	- / _	-	- B
ooo .	; / :	&	; CB 842
o oo. oo	* / |	$	* B 84
ooo. oo	, / <	,	, C A 8 21
oo o. oo	. / >	.	. BA 8 21
oo . o	/ / ?	/	/ C A 1
o o. o	Backspace	P13	@ A 8421
o .	Carr. Return	EOL	EOR CBA 8421
oooo.ooo	Tape Feed	TF	(data omit)
oooo.o	Upper Case	¤	= C A 84 1
oooo. o	Lower Case	SP1	! B 8 2
o .	Space	SP	SP C
ooo.oo	Tab Stop	SP	SP C

Appendix 4

Machine Configuration

IBM 1011 Paper Tape Reader

IBM 1401-03
- 8K Storage*
- Special features:
  - Indexing
  - High-Low-Equal Compare
  - Store Address Register Instructions

Friden Flexowriter Model SPD

* The program requires 3598 locations plus 2 additional storage areas for the character input and analysis. The current program assigns these areas at locations 4000 and 5000 respectively. If the assignments are made at 3600 and 3800 respectively (with the restriction of less than 200 input characters per record), a 4K storage 1401 computer is adequate.

Appendix 5

ALCOR Group 5-Channel Paper Tape Convention

The following is reproduced from Communications of the A.C.M. 6(1963), 598.

TABLE 1. 5-TRACK PAPER TAPE REPRESENTATIONS OF CHARACTERS USED FOR ALGOL

No.	Hole Combination: Track Feed hole 1 2 3 4 5	Designation by Track Numbers	Character Used for ALGOL: Letter Shift	Figure Shift
1	o o .	1 - 2	A	−
2	o . o o	1 - 4 - 5	B	× (multipl. operator)
3	o . o o	2 - 3 - 4	C	:
4	o . o	1 - 4	D	(not permitted, ▲)
5	o .	1	E	3
6	o . o o	1 - 3 - 4	F	[
7	o . o o	2 - 4 - 5	G	]
8	. o o	3 - 5	H	10 (ten)
9	o . o	2 - 3	I	8
10	o o . o	1 - 2 - 4	J	;
11	o o . o o	1 - 2 - 3 - 4	K	(
12	o . o	2 - 5	L	)
13	. o o o	3 - 4 - 5	M	. (point)
14	. o o	3 - 4	N	, (comma)
15	. o o	4 - 5	O	9
16	o . o o	2 - 3 - 5	P	0 (zero)
17	o o . o o	1 - 2 - 3 - 5	Q	1
18	o . o	2 - 4	R	4
19	o . o	1 - 3	S	' (apostrophe)
20	. o	5	T	5
21	o o . o	1 - 2 - 3	U	7
22	o . o o o	2 - 3 - 4 - 5	V	=
23	o o . o	1 - 2 - 5	W	2
24	o . o o o	1 - 3 - 4 - 5	X	/
25	o . o o	1 - 3 - 5	Y	6
26	o . o	1 - 5	Z	+
27	. o	4	(carriage return)	
28	o .	2	(line feed)	
29	o o . o o o	1 - 2 - 3 - 4 - 5	(letter shift)	
30	o o . o o	1 - 2 - 4 - 5	(figure shift)	
31	. o	3	(space)	
32	.		(not permitted)	(not permitted)